

# Team 38 Final Report

## 1. Project Idea

In this project, we are going to build a website called Movlen, which means we will use IMDb's database. Unlike IMDb, all functions in Movlen are closely related to movie directors. We want to create a website where information about movie directors can be stored and searched, news about movie directors can be published and talented directors can be discovered.

The main functionality of Movlen is to store and present information about movie directors. The functionalities of this web application is based on users. There are 2 types of potential users: ordinary user and verified directors. All users can search for any movie director. The webpage of a director will present his/her bio information, movie information, as well as some statistics and analysis of his movies, such as summary of genre and box office.

For different kinds of user, Movlen will also provide some additional functions:

- Ordinary user — has the right to follow and unfollow any movie director, public/private a post, like others' post; use Movlen's recommender system, through setting his/her only criterion, obtain personalized recommendation for movie director based on information including movie genre, box office earning, movie rating; they can also have right to experience the interactive explorer of movies, which is a visualization of overview of movie market;
- Verified director — is allowed to share news about his new movies on his webpage.

Among those functions, recommender system, visualization and full function social network are advanced functions.

## 2. ER Design and Schema Design

We assume that it's possible for two movies to have the same name, so do directors, actors and website users. So we assign a unique id to each movie, director, etc., and use it as key. We also assume that one movie might have two directors, so the relationship "direct" between director and movie is many to many.

### 2.1 ER Design

The ER diagram is showed in fig 2.1.1.

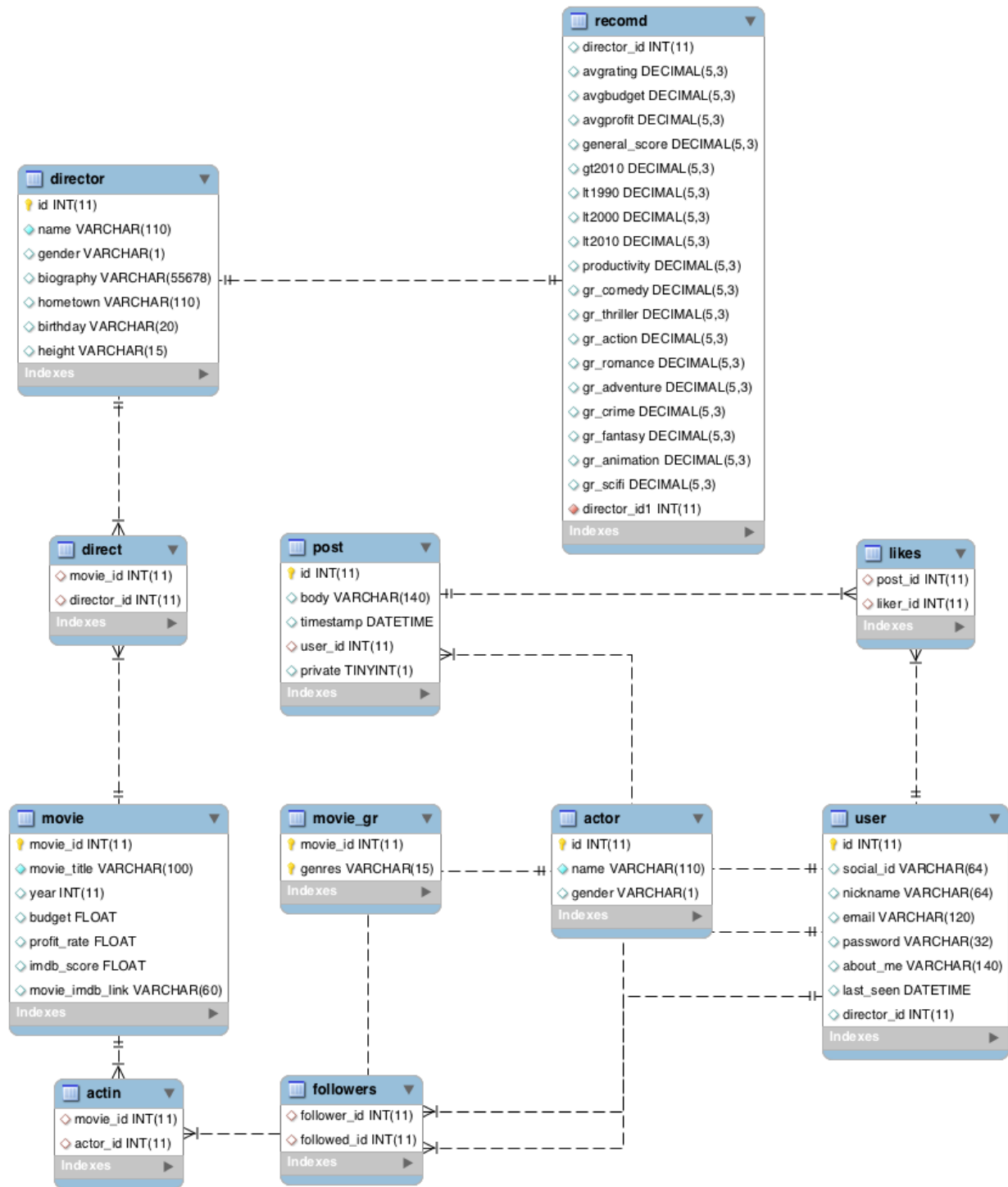


Fig 2.1.1 ER Diagram

## 2.2 Relational Schema

director entity:

(id, name, hometown, birthday, gender, biography, height)  
id → name, hometown, birthday, gender, biography, height

user entity:

(id, social\_id, nickname, email, password, about\_me, last\_seen, director\_id)  
id → social\_id, nickname, email, password, about\_me, last\_seen, director\_id

movie entity:

(movie\_id, movie\_title, year, budget, imdb\_score, profit\_rate, movie\_imdb\_link)  
movie\_id → movie\_title, year, budget, imdb\_score, profit\_rate, movie\_imdb\_link

movie\_gr entity:

(movie\_id, genres)

actor entity:

(id, name, gender)  
id → name, gender

post entity:

(id, timestamp, body, user\_id, private)  
id → timestamp, body, user\_id, private

actin relationship:

(actor\_id, movie\_id)

direct relationship:

(director\_id, movie\_id)

follow relationship:

(follower\_id, followed\_id)

likes relationship:

(post\_id, liker\_id)

recomd entity:

(director\_id, avgrating, avgbudget, avgprofit, general\_score, gt2010, it1990, it2000, it2010, productivity, gr\_comedy, gr\_thriller, gr\_action, gr\_romance, gr\_adventure, gr\_crime, gr\_fantasy, gr\_animation, gr\_scifi)

### 3. Data Import

#### 3.1 Data processing

We used the IMDb dataset as raw data set. It's a dataset of movies and related things that

can be found from this link <http://www.imdb.com/interfaces>. Its size is around 1 GB in total.

Then by using the tool IMDbPY, the raw data set is transferred into MySQL database. IMDbPY is a Python package useful to retrieve and manage the data of the IMDb movie database. After the transformation of IMDbPY, the database become around 15.7 GB in size and contains 21 tables. They can be divided into five main parts. One is title part, including the information about movie and cast information; the second is keyword, including movie keyword; third part is company\_type, including data of movie companies; the next part is kind\_type, including the information about movie kind type, like movie, TV series, episode etc.; the last part is info\_type, including person information of directors, actors, writers etc.

Next, we constructed our own database by extracting relative data according to our database schema. To achieve this, we write the SQL query to create 7 new tables. Then add the primary key in each table, adding foreign key if necessary. Like in the table actin, the movie\_id is the foreign key of the primary key movie\_id in table movie; the actor\_id is the foreign key of the primary key in the table actor. Besides, we made the primary key having the property of auto increment, that is, the value of the primary key field to be created automatically every time a new record is inserted. Also, we put constraint on the relationship of tables. Three triggers were created like once there is a delete on table movie, the corresponding tuples containing this movie\_id on tables actin, direct, movie\_gr, recomd (the table about director recommendation generated later) will be deleted. Likewise, two triggers are activated when there is a delete on tables director, actor.

### 3.2 Final data table

In the final database, there are totally 11 tables and two main part in the database. One is the movie data reorganized from IMDb database, the other part is data about user management, generating from our user information and their activities like they can follow director, public/private a post and they also can like others' post. In the movie data, there are 7 tables actin, actor, direct, director, movie, movie\_gr and recomd, covering the information about movie, director, actor and their relationships.

**Table 3.2.1 Final Data Tables Statistics**

	Movie part							User management part			
	actin	actor	direct	director	movie	movie_gr	recomd	user	followers	likes	post
Number of attributes	2	3	2	7	7	2	19	8	2	2	5
Number of rows	14716	6100	4257	1850	4919	14160	1850	7	13	4	20

We searched another IMDb data source from Kaggle website, and extracted the variables IMDb\_score and IMDb\_link from this data source, then join with our IMDb database. Since there are many missing values and extremely special characters in the IMDb database, to

guarantee the cleanness and completeness, we made a selection on the IMDb database and only remained the merged part. Thus, there is extremely small number of missing values in our database.

## **4. Development Environment**

### **4.1 Programming Language**

SQL: For operations in database

Python: Build control level (server behaviours) and model level (data operation in server) and some presentation level (through jinja) of the website

Java script: Introduce interactions to webpage

Html: Build up webpage

Jinja: Connector between python and webpage, which makes dynamic webpage possible

CSS: Design style sheet for webpage

### **4.2 Code Frameworks**

Flask: Python server frame work

Flask-sqlalchemy: Database connector and ORM for Flask, which is used to connect and operate database

Flask-whooshalchemy: Full text search support, to speed up the search in database

Bokeh: Data visualization package, which is used to perform data visualization

Rauth: Used to provide support for third party login

Bootstrap: Help improve website interface

D3: A java script based data visualization method

### **4.3 APIs**

Third party login support: The website supports third party account login, such as Yahoo!, AOL, Facebook;

Avatar embed: We use third party avatar website to provide avatar support for users.

### **4.4 Why we choose them**

Python is simple and quick to develop, SQL we have no choice, it is the same with html, css, java script, so I really do not know why should we talk about why we choose a language.

Flask: a swift server frame work of python, it is simple and quick

Bokeh: a good framework of python to build beautiful and interactive data visualization

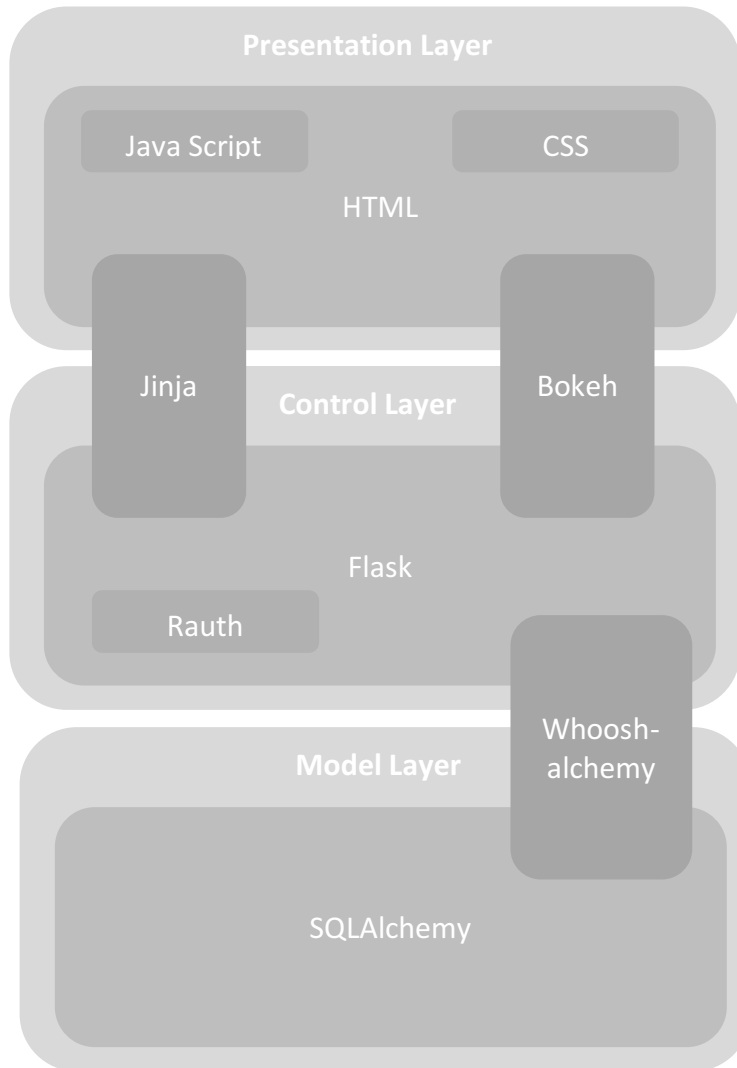
Rauth: a required framework of OAuth and OpenId

Bootstrap: it is a beautifully designed front end framework, which can help us improve the user interface

SQL-alchemy & whoosh: common used SQL connector and a fast full text search framework

## 4.5 Diagram

Diagram is shown as Fig 4.4.1.



**Fig 4.4.1 Development Environment Diagram**

## 4.6 Experience

There are many tricky things of the flask and data visualization, since all things are brand new and we have to work them out on our own.

## 5 Basic functionalities

### 5.1 Sign in, Log in and Log out:

Nickname + password login:

**Sign in**

Nickname:

Pssword:

Or login with:

| [Yahoo](#) | [AOL](#) | [Facebook](#)

☐ Remember Me

Don't have an account? [Create one!](#)

Third party account login (Yahoo!, AOL, Facebook) login:

**facebook**

Log into Facebook

[Forgot account?](#) - [Sign up for Facebook](#)

New user signing up:

**Sign Up**

Nickname:

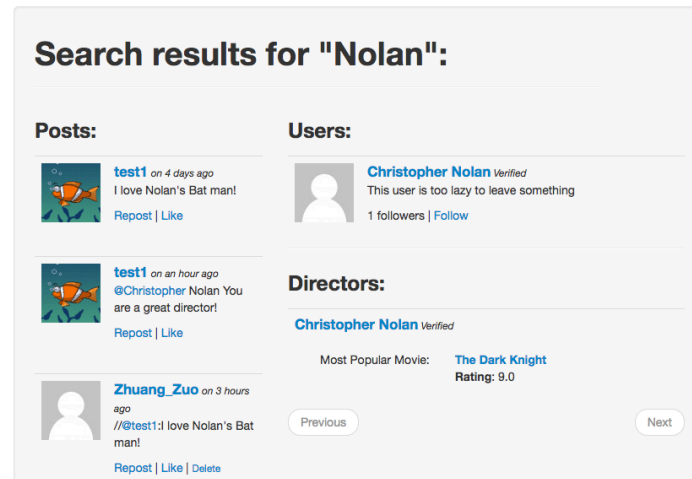
Email:

Password:

Re-enter password:

## 5.2 Full text search

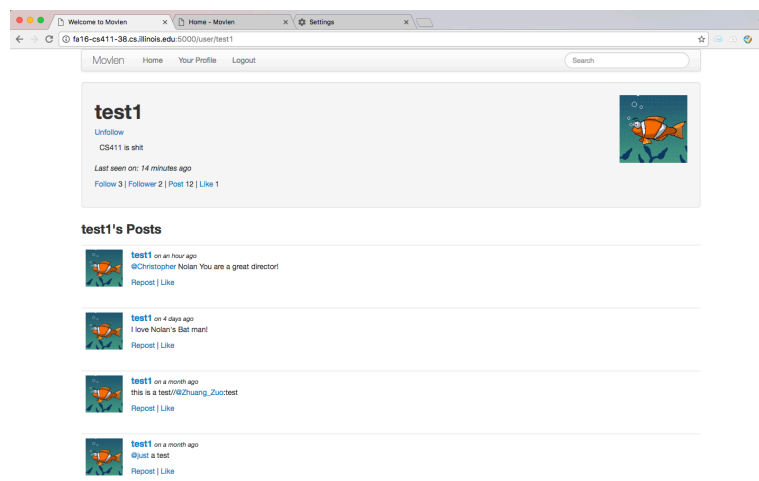
Our search function can search every contents of our website, from user to directors, as well as posts:



### 5.3 Fully interact website

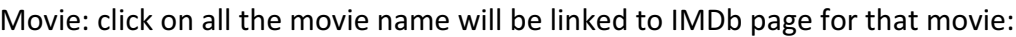
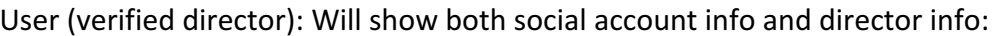
All the director name, user name and movie name are clickable, and each have different links.

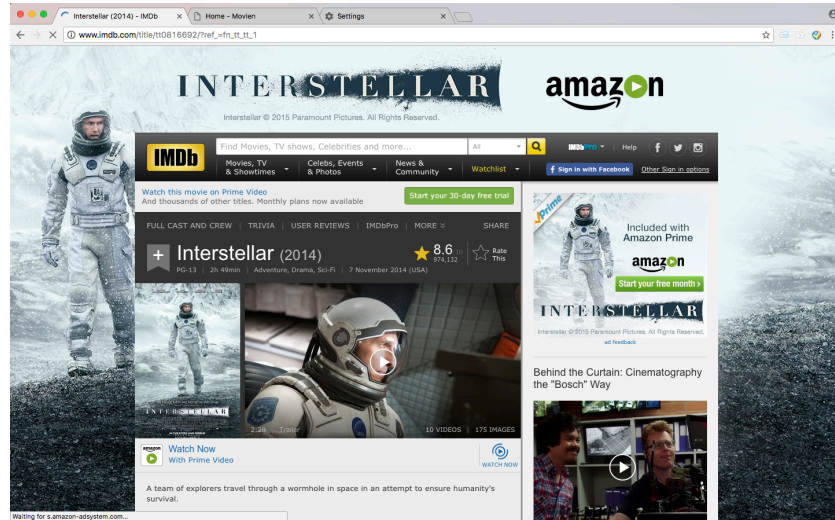
User (not a verified director): link to user's homepage, with user social account info and posts only:



director (not a user): Will Show director biography, director info, and movie related to this director only.

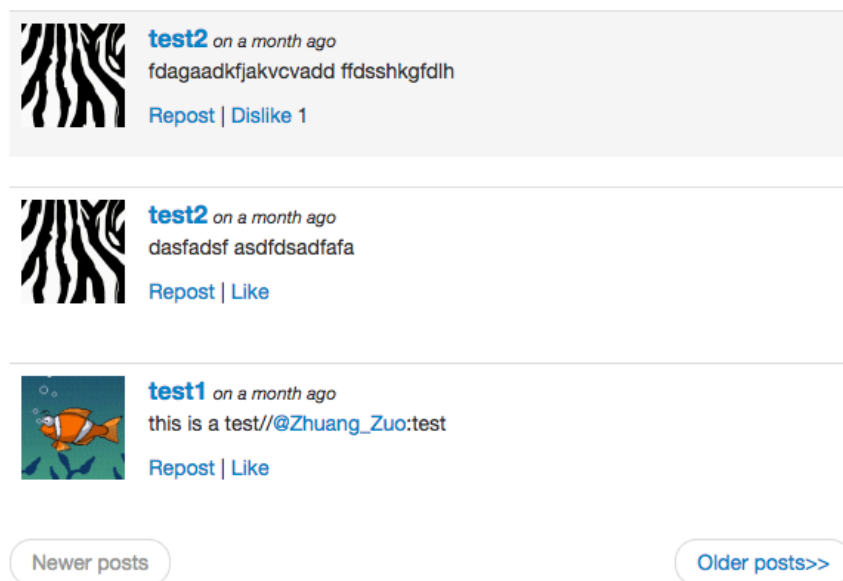






## 5.4 Page split

If the content is too much for a page to show (will cause the web page too long for a user to scroll down for a long time), we introduced previous / next page function:



## 6 Advanced functionalities

### 6.1 Recommender System

The recommender system is important for our website because it helps ordinary users find directors being good at certain areas, and it also helps movie producers, who might be more interested in finding a director who is good at making profitable movies.

We offer users multiple choices of recommending criterion, including:

Genre – number of movies directed for a certain genre

Movie Rating – average movie rating on IMDB

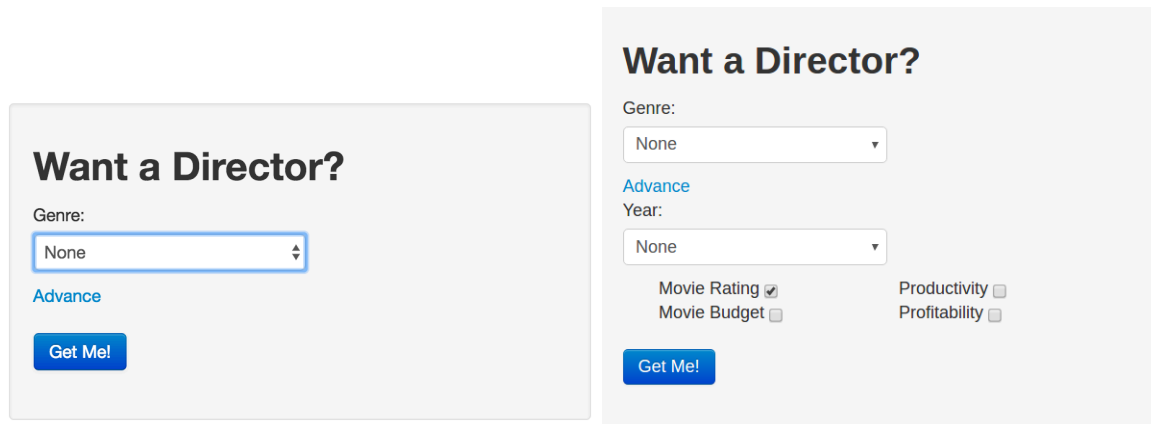
Year – number of movies directed in a certain time period

Productivity – total number of movies having directed

Movie Budget – average budget of movies

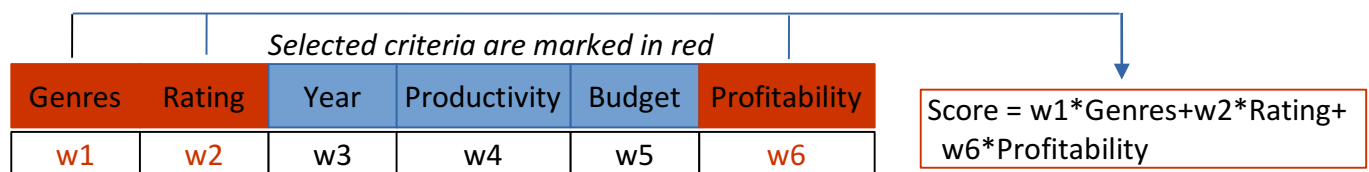
Profitability – average proportion of box office earnings to budget

We think Genres and Rating are the most important factors taken into account when looking for a director, so in our recommender system, Genres is shown by default, and Rating is by default selected (in “Advance”). As shown in Fig 6.1.1, users can click on “Advance” to see and select other criteria.



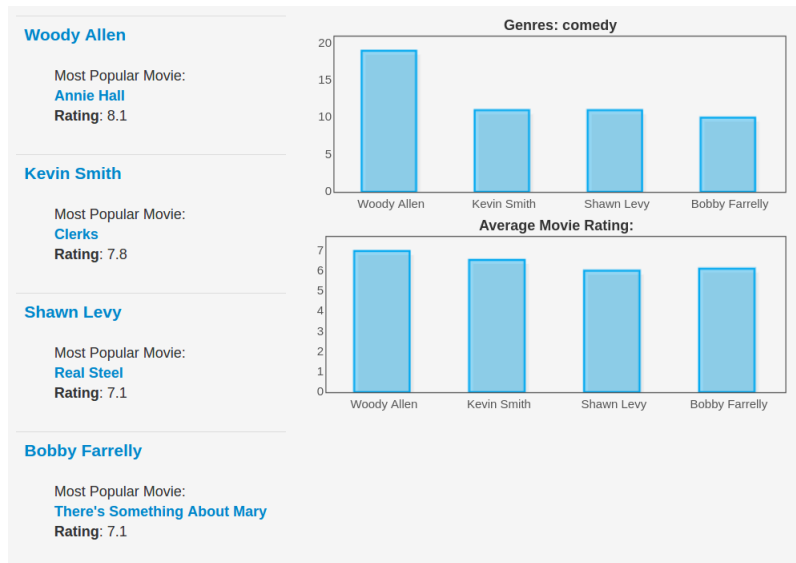
**Fig 6.1.1. Criteria for director recommendation system**

Once a user clicks “Get Me!”, the back end will get the information about which criterion gets selected and which doesn’t. The recommender system will calculate a score for each director based on the selected criteria and return the name for the top 4 directors. The recommendation algorithm is shown in Fig 6.1.2. The final score will be the weighted average of the sub-scores chosen by the user. And the weights are adjusted that each sub-score will have similar contribution to the final score.



**Fig 6.1.2. Recommendation algorithm**

Kevin is a comedy movie fan, and he wants to use our recommender system to find a director good at comedy movie. By selecting “Comedy” in Genres and “Movie Rating” (this is selected by default), he will get a recommendation web-page looks like Fig 6.1.3.



**Fig 6.1.3. Recommendation web-page example**

Compared with other recommender system, our recommender system will not only present the names of recommended directors, it will also explain the reason why the recommendation is made. From the bar plots, Kevin will see Woody Allen is on the top of the list because he has directed a lot of comedy movies and the average rating of his movies is also very high, so Kevin will be more convinced by the recommendation results.

Zack is more interested in Adventure movie, after selecting “Adventure” in Genres and check all other criteria, he will get the following recommendation web-page, which reasonably shows Steven Spielberg on the top. So our recommendation system is flexible in providing different recommendations based on different requirements and more interpretable in presenting bar plots to explain the reason to make the recommendation.

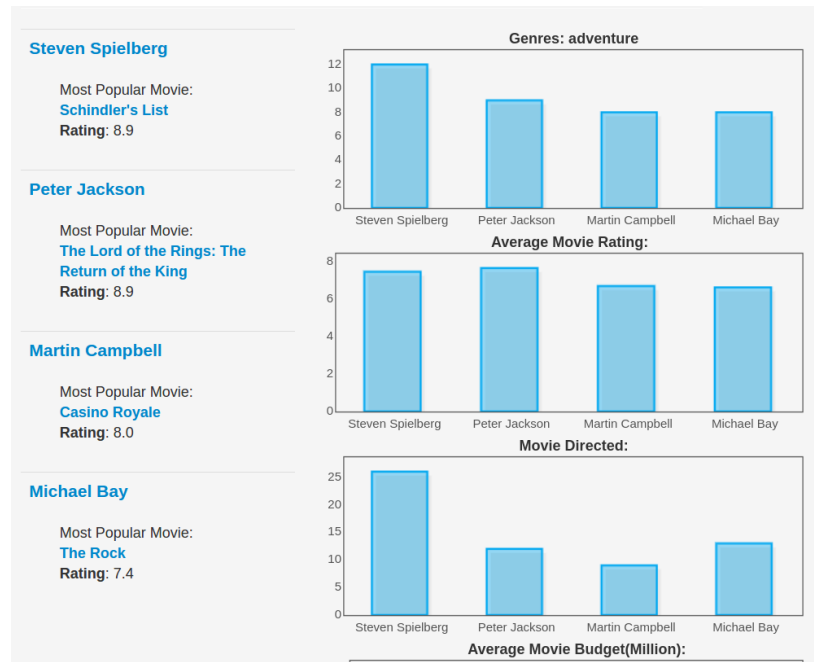


Fig 6.1.4. Recommendation for director good at adventure movie

## 6.2 Advanced Visualization

Our visualization tool is an interactive explorer of IMDb movie market. It is a nice tool for directors and producers. Based on the designing principle that users love select and hate enter, we provide sliders and drop down boxes for users to set minimum movie scores, year windows and genres of movies.

In the plot, movie points are labeled different color, size and transparency. When hovering over those points, more detailed information about the movie is demonstrated. The x and y axes auto scale as users zoom in or zoom out. People can even save the plot after they set selection conditions.

### AN INTERACTIVE EXPLORER FOR IMDB DATA

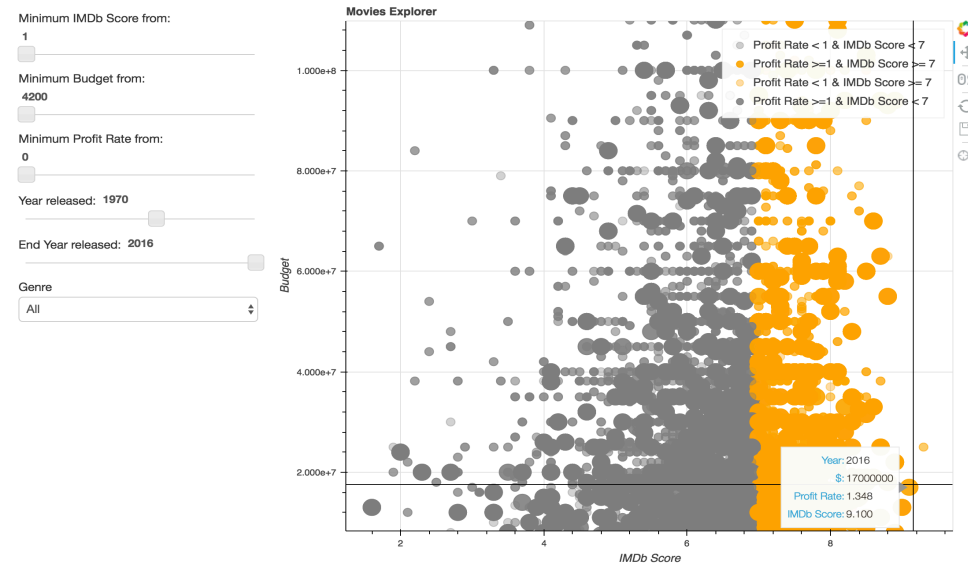


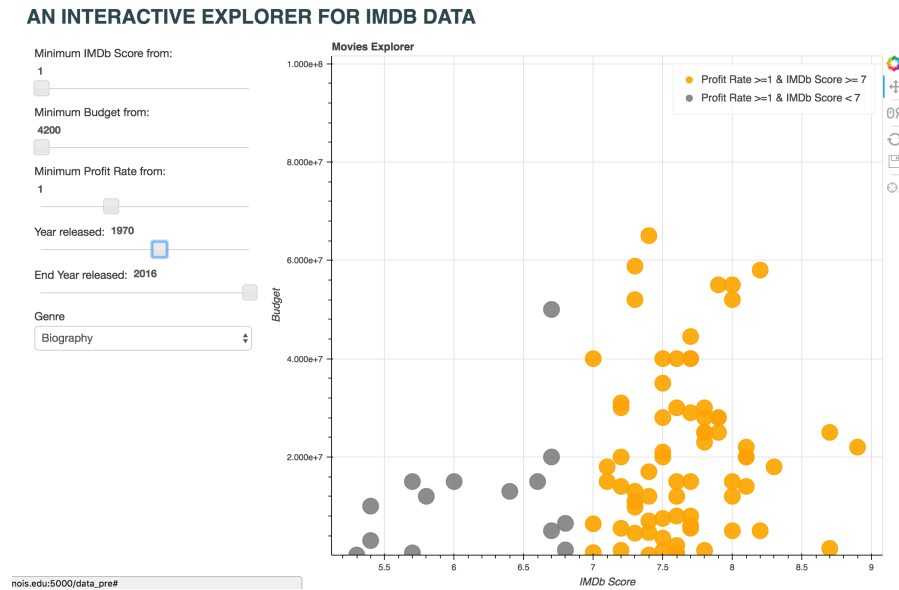
Fig 6.2.1 Interactive Explorer for IMDB data

#### 6.2.1 Importance

It helps movie-lovers, directors and producers understand the change of the movie market as time goes by, check the budget, score and profit rate of different genres of movies.

Say Kevin is a producer and he wants to make an action movie but has no general idea if it requires huge budget to achieve high IMDb score. Then he can select "Action" in the genre drop-down box and see from the plot that most action movies do need more budget. He can also move the year slider and view changes of movie market through years.

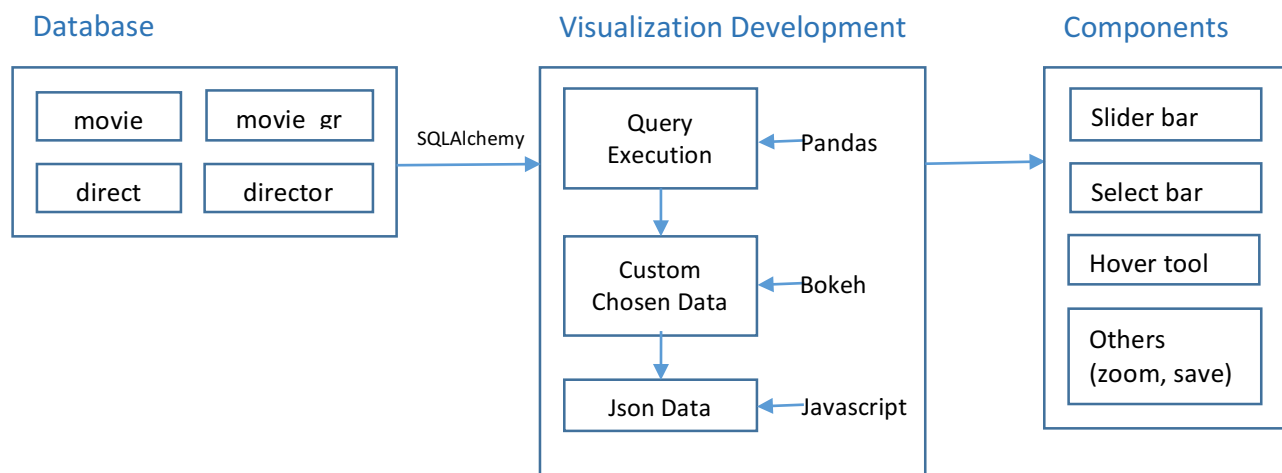
For director Joe, he can use our tool to know the popularity of different genres of movies and direct movies whose genres cater to the market. Suppose Joe is not well famous yet prefers low budget movies that with high score, he can combine selection and slider tools to view the market. For example, movies in biography does not have high budget but high proportions of them gets both high profit rate and high IMDb scores. Joe possibly can choose to direct biography as his next movie.



**Fig 6.2.2 Biography Example**

## 6.2.2 Solution

**Table 6.2.1 Visualization Data Flow**



### 1) Database Connection

We used SQLAlchemy in Python to create database engine and connect to database microblogdb hosted in our server. Then wrote SQL query to join movie, movie\_gr (movie genre), direct and director tables to get movie\_id, movie\_title, year, budget, profit\_rate, imdb\_score and director information.

### 2) Visualization Tool Development

We realized slider bar and dropdown list as main function by combining Python libraries and JavaScript.

The visualization is constructed using Python Bokeh library, which provides interactive ways to illustrate data. Normally Bokeh plots are shown in Bokeh server, we did huge modifications to incorporate real-time data and load plots in our own server.

For graph demonstration, based on the data returned from database queries, we used Python Pandas to generate a dataframe and labeled movies in different color and transparency according to IMDB score and profit rate.

For graph interaction, we programmed in JavaScript and Python to callback data update in Bokeh plotting procedure to filtered out movies that satisfy selection conditions.

### 3) Webpage Embedding

Our data are transformed to Json format internally; our plots are saved as JavaScript script and HTML div components manually and got passed to a HTML webpage that demonstrates the plot as variables. Jinja framework is applied in HTML design so HTML elements in visualization page are written in Jinja style as well. JavaScript and CSS related to Bokeh plotting are meanwhile passed to the HTML file.

#### 6.2.3 Evaluation

Advanced movie visualization tool not only offers fancy search functions, but also guarantees easiness to use and nice-looking of the output. Furthermore, we provide zoom in/out and save functions, make it possible for users to speculate on certain points and save their search for future use. We evaluate advanced tool in the following perspectives, as shown in the table.

**Table 6.2.2 Evaluation for Advanced Movie Visualization**

Dimension	Detail
Function	The slider and drop-down box are powerful and in combination lead to varieties of selections.
Interaction	Real-time data is shown and get updated quickly.
Appearance	Good-looking interface, colorful points with different size.
Convenience	Easy to use. No need to give extra instructions for users to play with this tool.
Creativeness	No other app provides such plots. Information about movie release year, budget, IMDB score and genre needs to be searched one movie at a time and users usually get text returns instead of plots.

### 6.3 Full Function Microblog Network

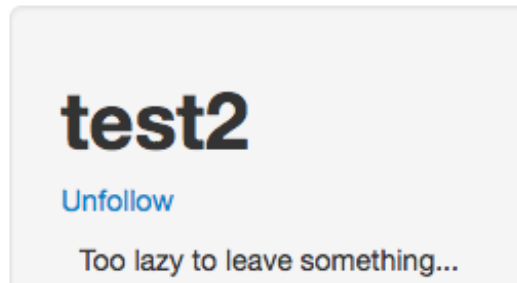
#### 6.3.1 Importance

People are social creature, which means social network is import to people in each area. With this function, directors and their fans can stick together and exchange their thoughts and lives.



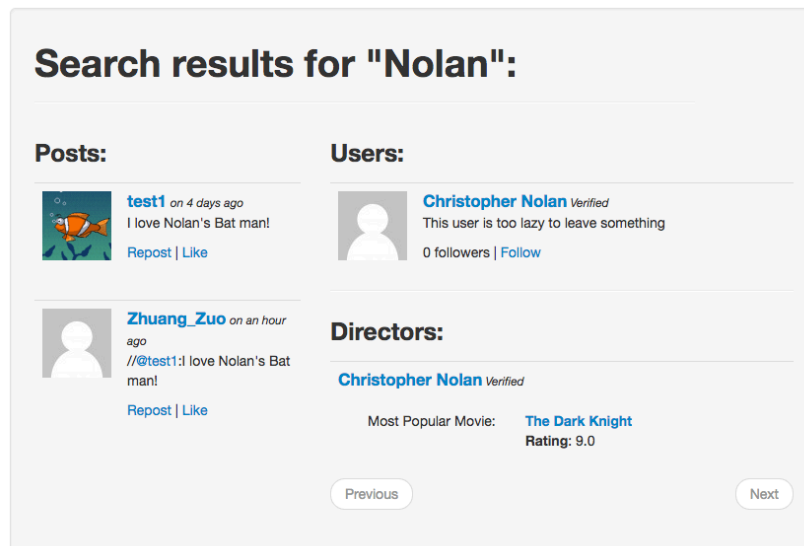
## 6.3.2 Components

### 6.3.2.1 User can follow and unfollow other users:

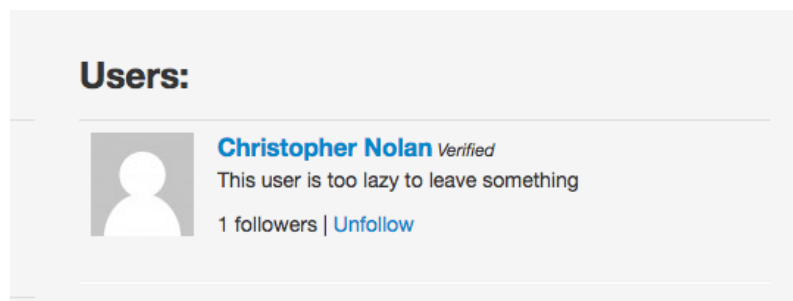


### 6.3.2.2 Verified Users

For example, Kevin likes Christopher Nolan (director) very much, he wants to stay connected with Nolan(director), he can then search Nolan's name, and the result shows as follow:



From the result we can see there is a verified user Christopher Nolan, which means this user is a real director that are verified by Movien. Then Kevin can click “follow” to get access to Nolan's posts. After that, he can click on unfollow to unfollow Nolan



### 6.3.2.3 Public post / Private Post / delete

If a user wants to keep his / her post private, he /she can choose “Myself” while posting:

## Hello, Christopher Nolan!

What's going on?

This website's server works like shit

show this post to

Myself

Post

And this post will not appear in other's timeline.

## Hello, test1!

What's going on?

show this post to

Public

Post



**Christopher Nolan** on 28 minutes ago  
Come and have a look at my movie - Bat man!  
[Repost](#) | [Like](#)



**test1** on 4 days ago  
I love Nolan's Bat man!  
[Repost](#) | [Dislike 1](#) | [Delete](#)



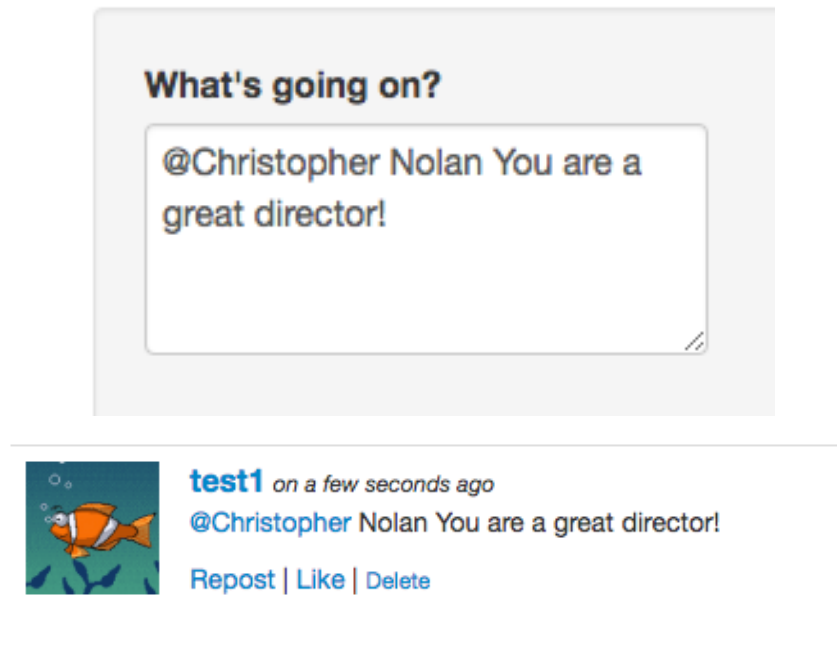
**test2** on 5 days ago  
this is a test for Public  
[Repost](#) | [Like](#)

If you want to delete an unwilling post you can click delete to remove it.

### 6.3.2.4 Mention (@), Repost and like

You can mention other users by adding “@” symbol between their user name, and the

“@nickname” will show up as a link to that user’s info page in the post.



If you find other’s posts interesting, you can repost it, and the author of the post will automatically be mentioned:

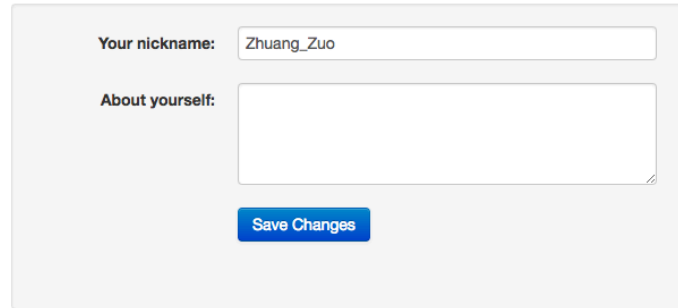


If you like a user’s post, you can click “like” bellow to show your affection.

#### 6.3.2.5 User profile edit

User can edit their profile:

## Edit Your Profile

A web form titled "Edit Your Profile" with a light gray background. It contains two input fields: "Your nickname:" with the text "Zhuang\_Zuo" and "About yourself:" with a larger empty text area. Below the fields is a blue button labeled "Save Changes".

Your nickname:

About yourself:

### 6.3.3 Evaluation

This function is a quite good function, which adds the interaction among users of the website, which can add up user loyalty and activity user rate.

### 6.3.4 Sufficient effort

Have to learn Regular expressions to detect “mention (@)” operations in users’ posts, and then pass a link from server to front end webpages.

When user commit a repost behavior, server have to capture the original post and automatically add mention(@) author of the post.

### 6.3.5 Creativity and Advanced points

No other group thought about build up a social network for their user, but only provide services. Few group can realize the function like “mention”.

## 7 Future work

In visualization part, for users to locate statistics of a specific movie and view plots of movies of a specific director, we will make each point in the plot more recognizable and add function to show movies made by directors that users take interest in.

When hovering over the movie, the movie title can be seen so that people know exactly the rating, budget, profit rate, etc. of a certain movie. Moreover, a link to the IMDb movie page should be activated when clicking on the movie point in our plot.

Besides, a form will be added for users to enter director name. Returned movies are all related to that director. Text input contents can be vague, which are case insensitive. Directors whose names contain the input strings will all be shown.

## 8 Labor division

Zhuang Zuo, Zhoutao Pei: Back-end, recommendation system

Yuetong Liu, Rong Du: Front-end design, interactive data visualization